

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of aiding deployment in a distributed computer system, using application software components, comprising:

[[a.]] (a) providing a tree representation (~~510~~) of objects existing in at least one of the application software components, the tree representation including an object as a leaf node, with at least one attribute of the object being in nexus between the root and the leaf node, wherein the tree representation is mapped to a containment tree that defines names of the attributes in the tree representation, and

[[b.]] (b) providing an accessor object (~~511~~), comprising accessor methods to the tree representation, said accessor methods having predefined names and functions.

2. (Currently Amended) The method of claim 1, wherein: stage [[a.]] (a) comprises providing a tree representation (~~510~~) in which at least one of the tree nexus comprises data defining both an attribute name and an attribute value.

3. (Currently Amended) The method of claim 2, wherein stage [[b.]] (b) comprises: [[b1.]] (b1) converting the tree representation (~~510~~) into at least one set of data (E2-8; E2-9) connecting an attribute with a corresponding location in the tree, and [[b2.]] (b2) storing at least one set of data or a designation thereof in the accessor object (~~511~~).

4. (Currently Amended) The method of claim 3, wherein stage [[b1.]] (b1) comprises converting the tree representation into a set of data (~~E2-8~~) connecting an attribute name with a corresponding location in the tree.

5. (Currently Amended) The method of claim 3, wherein stage [[b1.]] (b1) comprises converting the tree representation into a set of data (~~E2-9~~) connecting an attribute value with a corresponding location in the tree.

6. (Currently Amended) The method of claim 4, wherein stage ~~[[b1.]]~~ (b1) comprises converting the tree representation into a set of data ~~(E2-9)~~ connecting an attribute value with a corresponding location in the tree.

7. (Currently Amended) The method as claimed in ~~any of~~ claim 1, wherein the attributes are arranged in the tree in accordance with a predefined order.

8. (Currently Amended) The method as claimed in ~~any of~~ claim 1, wherein stage ~~[[b.]]~~ (b) comprises providing a plurality of accessor objects ~~(511,521)~~, with each accessor object comprising a domain identification, being readable through at least one ~~(M1)~~ of the accessor methods.

9. (Currently Amended) The method of claim 8, wherein at least two of the accessor objects ~~(511,521)~~ access different sub-trees in the tree representation.

10. (Currently Amended) The method as claimed in any of claim 8, further comprising the step of: ~~[[c.]]~~ (c) providing one or more handler objects ~~(611,612)~~, each accessing at least one of the accessor objects ~~(511,521)~~.

11. (Currently Amended) The method of claim 10, wherein stage ~~[[c.]]~~ (c) comprises: ~~[[c1.]]~~ (c1) configuring the handler objects ~~(611,612)~~ to access accessor objects ~~(511,521)~~ corresponding to application software components being currently in service in the distributed computer system.

12. (Currently Amended) The method of claim 11, wherein stage ~~[[c1.]]~~ (c1) uses attributes of the application software components.

13. (Currently Amended) The method of claim 10, further comprising the stage of: ~~[[d.]]~~ (d) in at least one client software application, providing a lookup service ~~(711)~~, adapted to communicate with at least one handler object ~~(612)~~.

14. (Currently Amended) The method as claimed in any of claim 1, wherein said accessor methods of stage ~~[[b.]]~~ (b) comprise a method to retrieve supported attribute names.

15. (Currently Amended) The method as claimed in any of claim 1, wherein said accessor methods of stage ~~[[b.]]~~ (b) comprise a method for exact search in the tree representation.

16. (Currently Amended) The method as claimed in any of claim 1, wherein said accessor methods of stage ~~[[b.]]~~ (b) comprise a method for near search in the tree representation.

17. (Currently Amended) The method as claimed in ~~any of~~ claim 1, wherein said accessor methods of stage ~~[[b.]]~~ (b) comprise a method capable of navigating up in the tree to find a leaf node.

18. (Currently Amended) An article of manufacture for reorganizing data in an original log file, the article comprising:

data defining a tree representation (510) of objects existing in at least one application software component, the tree representation including an object as a leaf node, with the attributes of the object being in nexus between the root and the leaf node, wherein the tree representation is mapped to a containment tree that defines names of the attributes in the tree representation, and

first tree access code (511) attaching accessor methods to the tree representation, said accessor methods having predefined names and functions.

19. (Currently Amended) The article of claim 18, wherein said first tree access code comprises an accessor object ~~(511)~~, comprising the accessor methods.

20. (Currently Amended) The article of claim 17, wherein the accessor object ~~(511)~~ has a predefined name.

21. (Original) The article of claim 17 further comprising: a second tree access code, adapted to access the first tree access code.

22. (Currently Amended) The article of claim 21, wherein said second tree access code comprises a handler object ~~(612)~~.

23. (Currently Amended) The article of claim 21 further comprising: a third code for incorporation in at least one client software application, said third code providing a lookup service ~~(711)~~, adapted to communicate with the second tree access code.

AMENDMENTS TO THE DRAWINGS

A replacement sheet of Figure 1 is filed herewith, wherein Figure 1 is amended to add the legend "Prior Art".